Introduction to Programming II

Assignment 2: Scheduling

Deadline: Week 14 (23.04.19 23:59)

Output: Only one C file with the code has to be uploaded to Moodle. Input, output, header and other files are NOT allowed. Name of the file should be like this *NameSurname.c* (For example, *IvanIvanov.c*). No other symbols allowed

Programming language: C (C11 standard)

Requests:

- The program must work, the code should be readable, well-structured and should contain English comments
- It has to be only one **.c* file and nothing else
- It is allowed to use only standard C libraries
- NO extension of a deadline. Works sent after the deadline will NOT be evaluated
- Assignment is individual
- We will be using MOSS (Measure of Software Similarity) as a test for plagiarism. Be reminded that a score of 0 will be assigned to any submissions suspected of plagiarism pending a full investigation as per IU policies.

Evaluation criteria: 80% for the code correctness, 20% for readability of code and comments

Task:

Every year the Dean must suffer through the process of putting professors and TAs against classes. He has asked for a new system to be developed which will simplify the process.

A class has a single-word name, number of students allowed (at least 1), several Labs which have a TAs required (at least 1), and a professor. If a course is missing a professor or a TA assigned to a lab then it will not be run. A class will appear in this list if it is trainable (by professors or TAs) or selectable by students. Class name can contain only English letters.

A student has a first and last name, a Student ID, and at least 1 selected class. Student's first and second name can contain only English letters. Student ID can contain English letters and numbers (exactly 5 symbols).

A Professor has a first and last name and at least 1 course for which he is trained. A professor can be assigned to at most two classes for which they are trained or a single class for which they are not trained. Professor's first and second name can contain only English letters.

A TA has a first and last name, and at least 1 trained course. TAs may be assigned to up to four classes (including taking multiple labs in the same course) but may not be assigned to any class for which they are not trained. TA's first and second name can contain only English letters.

Inputs: Input should be represented by several files called *inputN.txt* (For example, *input1.txt*), where *N* is number of current input file. Value of *N* should begin with 1 and end with at most 50 (can be less). Input files should be in root with your **.c* file. Each input file will be structured as follows:

All Course Name, followed by the number of required labs, followed by the number of allowed students, space delineated the list terminated by a P on a new line
Professor's First and Last Name followed by all classes he is trained for, space delineated the list terminated by a T on a new line
TA's First and Last Name followed by all classes he is trained for, space delineated the list terminated by an S on a new line
Student's First and Last Name followed by Student ID, followed by all their requested classes, space delineated

Example:

Math 2 50
Programming 3 70
P
Joseph Brown Math Programming
Giancarlo Succi Programming
T
Munir Makhmutov Programming
Marat Mingazov Math
S
Mark Lancaster 02930 Math Programming
John Smith 92383 Programming

Names, surnames and course names cannot be P, T and S. Each person can have only one role: Professor, TA or Student. Hypothetically, a Professor Joseph Brown, a TA Joseph Brown and a Student Joseph Brown are three different people inside one valid input.

Outputs: The Dean requires a system which will produce the least amount of problems in given EXACT order and gives the following scores to the items:

20 badness points if a course cannot be run.

```
[Course name] cannot be run.
```

10 badness points for a Professor who is unassigned to any courses.

`[Professor's first name and last name] is unassigned.`

5 badness points for a Professor who is working on a course for which they are not trained or if they are working on a single course.

`[Professor's first name and last name] is not trained for [course name].`

`[Professor's first name and last name] is lacking class.`

2 badness points for each of the 4 classes that TA is not assigned against.

`[TA's first name and last name] is lacking [x] lab(s).`

1 badness point for a student who cannot take a class on their requested list.

`[Student's first name and last name] is lacking [course name].`

Your system will search for a least bad solution and print out the following records in output files called like *NameSurnameOutputN.txt* (For example, *IvanIvanovOutput1.txt*), where N is number of current output file which corresponds to *inputN.txt* file. Value of *N* should begin with 1 and end with at most 50 (can be less). Output files should be in the root with your **.c* file. If input file contains wrong symbols, insignificant zeros (Student ID is exception) or wrong content, then output file should contain message: *Invalid input*. Each output file should have the next structure:

All courses and students have to be in the order of their occurrence in input file. Each Course list should contain:

Course Name
Assigned Professor's First and Last Name
Assigned TA(s) First and Last Name
Student List [Student's First and Last Name followed by a student number]

A List of all bad events, if there are any.
The total badness score.

Example:

Math
Joseph Brown
Marat Mingazov
Mark Lancaster 02930

Programming
Giancarlo Succi
Munir Makhmutov
Mark Lancaster 02930
John Smith 92383

Joseph Brown is lacking class.
Giancarlo Succi is lacking class.
Marat Mingazov is lacking 2 lab(s).
Munir Makhmutov is lacking 1 lab(s).
Total score is 16.

One more output file *NameSurnameEmail.txt (For example, IvanIvanovEmail.txt)* should be generated with the only string, which is your Innopolis email linked to Moodle. So, for each *N* input files there should be *N+1* output files.

Tests:

1. **Test 1**

   Input:
   Math 4 50
   Programming 4 60
   Philosophy 4 70
   P
   Joseph Brown Math Programming
   Giancarlo Succi Programming
   T
   Munir Makhmutov Programming
   Marat Mingazov Math
   Daniel Carvalho Philosophy
   S
   Mark Lancaster 02930 Math Programming
   John Smith 92383 Programming Philosophy
   Andy Turner 23498 Math Programming Philosophy
   Peter Myles 29398 Math Programming Philosophy
   Samuel Cage 69943 Programming Philosophy

   Output:
   Math
   Joseph Brown
   Marat Mingazov
   Mark Lancaster 02930
   Andy Turner 23498
   Peter Myles 29398

   Programming
   Joseph Brown
   Munir Makhmutov
   Mark Lancaster 02930
   John Smith 92383
   Andy Turner 23498
   Peter Myles 29398
   Samuel Cage 69943

   Philosophy
   Giancarlo Succi
   Daniel Carvalho

John Smith 92383
Andy Turner 23498
Peter Myles 29398
Samuel Cage 69943

Giancarlo Succi is not trained for Philosophy.
Total score is 5.

2. **Test 2**

Input:
Math 4 60
Programming 4 50
P
Joseph Brown Math Programming
T
Munir Makhmutov Programming
Marat Mingazov Math
S
Mark Lancaster 02930 Math Programming
John Smith 92383 Programming

Output:
Math
Joseph Brown
Marat Mingazov
Mark Lancaster 02930

Programming
Joseph Brown
Munir Makhmutov
Mark Lancaster 02930
John Smith 92383

Total score is 0.

3. **Test 3**

Input:

Math 4 20
Programming 4 20
P
Joseph Brown Math Programming
T
Munir Makhmutov Programming
Marat Mingazov Math
S
Mark Lancaster 02930 Programming
John Smith 92383 Programming

Output:

Math
Joseph Brown
Marat Mingazov

Programming
Joseph Brown
Munir Makhmutov
Mark Lancaster 02930
John Smith 92383

Total score is 0.

4. **Test 4**

Input:

Math 4 4
Programming 4 4
P
Joseph Brown Math Programming
T
Munir Makhmutov Programming
Marat Mingazov Math
S
Mark Lancaster 02930 Math Programming
John Smith 92383 Programming
Andy Turner 23498 Math Programming
Peter Myles 29398 Math Programming
Samuel Cage 69943 Math Programming
Ivan Ivanov 94563 Math Programming

Output:

Math
Joseph Brown
Marat Mingazov
Mark Lancaster 02930
Andy Turner 23498
Peter Myles 29398
Samuel Cage 69943

Programming
Joseph Brown
Munir Makhmutov
Mark Lancaster 02930
John Smith 92383
Andy Turner 23498
Peter Myles 29398

Samuel Cage is lacking Programming.
Ivan Ivanov is lacking Math.
Ivan Ivanov is lacking Programming.
Total score is 3.

5. **Test 5**

Input:

Math 4 40
Programming 4 40
P
Joseph Brown Math Programming
Giancarlo Succi Math Programming
Manuel Mazzara Programming
T
Munir Makhmutov Programming
Marat Mingazov Math
S
Mark Lancaster A2930 Math Programming
John Smith 92383 Programming
Andy Turner 23498 Math Programming
Peter Myles 29398 Math Programming
Samuel Cage 69943 Math Programming
Ivan Ivanov 94563 Programming

Output:

Math
Joseph Brown
Marat Mingazov
Mark Lancaster A2930
Andy Turner 23498
Peter Myles 29398
Samuel Cage 69943

Programming
Giancarlo Succi
Munir Makhmutov
Mark Lancaster A2930
John Smith 92383
Andy Turner 23498
Peter Myles 29398
Samuel Cage 69943
Ivan Ivanov 94563

Manuel Mazzara is unassigned.
Total score is 10.

6. **Test 6**

Input:

Math 4 40
Programming 4 40
Philosophy 4 40
P
Joseph Brown Math Programming
Giancarlo Succi Math Programming
Manuel Mazzara Philosophy
T
Munir Makhmutov Programming
Marat Mingazov Math
S
Mark Lancaster 02930 Math Programming
John Smith 92383 Programming
Andy Turner 23498 Math Programming
Peter Myles 29398 Math Programming
Samuel Cage 69943 Math Programming

Output:

Math
Joseph Brown
Marat Mingazov
Mark Lancaster 02930
Andy Turner 23498
Peter Myles 29398
Samuel Cage 69943

Programming
Giancarlo Succi
Munir Makhmutov
Mark Lancaster 02930
John Smith 92383
Andy Turner 23498
Peter Myles 29398
Samuel Cage 69943

Philosophy cannot be run.
Total score is 20.

7. **Test 7**

Input:

Math 4 40
Programming 4 40
Philosophy 4 40
P
Joseph Brown Math Programming
Giancarlo Succi Math Programming
Manuel Mazzara Philosophy
T
Munir Makhmutov Programming
Marat Mingazov Math
S
Mark Lancaster 02930 Math Programming
John Smith 92383 Programming Philosophy
Andy Turner 23498 Math Programming
Peter Myles 29398 Math Programming

Output:

Math
Joseph Brown
Marat Mingazov
Mark Lancaster 02930
Andy Turner 23498
Peter Myles 29398

Programming
Giancarlo Succi
Munir Makhmutov
Mark Lancaster 02930
John Smith 92383
Andy Turner 23498
Peter Myles 29398

Philosophy cannot be run.
John Smith is lacking Philosophy
Total score is 21.

8. **Test 8**

Input:

Math 4 40
Programming 4 40
P
Joseph Brown Math Programming
Giancarlo Succi Math Programming
Manuel Mazzara Philosophy *// this subject does not exist in the list of courses*
T
Munir Makhmutov Programming
Marat Mingazov Math
Daniel Carvalho Philosophy
S
Mark Lancaster 02930 Math Programming Philosophy
John Smith 92383 Programming
Andy Turner 23498 Math Programming
Peter Myles 29398 Math Programming
Samuel Cage 69943 Math Programming

Output:

Invalid input. *// other mistakes should not appear*